

**IN THE MATTER OF the Patents Act 1953**

**AND**

**IN THE MATTER of New Zealand Patent No.  
525484 ("the Subject Application") by the  
**MICROSOFT CORPORATION** ("the Applicant").**

**AND**

**IN THE MATTER of a (s 21) Opposition thereto  
by the **NEW ZEALAND OPEN SOURCE  
SOCIETY** ("the Opponent").**

---

**STATEMENT OF CASE**

---

-

JAMES & WELLS  
PRIVATE BAG 11907  
DX CP34005  
ELLERSLIE  
AUCKLAND  
PHONE: 09-914.6740  
FAX: 09-914.6794

SOLICITOR OF RECORD: D MARRIOTT

## STATEMENT OF CASE

### LOCUS STANDI – INTEREST TO THE OPPONENT

1. The New Zealand Open Source Society Incorporated (“the Opponent”) is a society incorporated under the laws of New Zealand, having its registered office at Unit A6, 400 Rosedale Road, Albany, Auckland, New Zealand.
2. The Opponent represents the interests of a number of commercial entities active in the development, manufacture and distribution of software solutions who are specifically lending their name to this opposition (see Appendix 17). Such activities include the manipulation of xml data from a single word-processing document.
3. The Opponent is primarily an advocate and promoter of open source software for business, private, education and government use.
4. The proposed grant of Letters Patent for Application No 525484 directly contradicts the primary objectives of the Opponent. Further, the proposed grant would be detrimental to the current and future activities of those commercial entities which the Opponent represents.
5. The Opponent therefore contends it has locus in these proceedings, and will adduce evidence in support (if required).

### NEW ZEALAND PATENT APPLICATION NO 525484

6. New Zealand Patent Application No 525484 (“the Subject Application”) relates to the implementation of a standard universal language, Extensible Mark-up Language (XML), in a single word-processing document file so as to provide non-native computer applications and services the ability to reproduce the rich formatting of the document as if it were one of the documents produced by the native application. Conversely, the publication of the schema of the native word-processor and XML file allows for the production by non-native applications of files understandable by the native application. By the use of this schema the word-processor is able to validate the word-processing document. By the implementation of hints within XML files non-native applications that understand XML are able to understand some of the features provided by the native word-processor without its full schema.

7. The Subject Application was filed as a Convention Application in reliance of a United States (US) patent filing dated 28<sup>th</sup> June 2002, US Application Number 187060.
8. The Subject Application was filed on 24<sup>th</sup> April 2003 claiming priority from US patent filing date.
9. The Subject Application was advertised as accepted on 20<sup>th</sup> January 2005 in New Zealand Patent Office Journal Number 1508 and published 25 February 2005.
10. The Subject Application currently reads as follows:

## Claims

1. A computer-readable medium having computer-executable components, comprising:
  - a first component for reading a word processor document stored as a single XML file;
  - a second component that utilises an XSD [XML Schema Document – schema] for interpreting the word-processor document, and
  - a third component for performing an action on the word-processor document.
2. The computer-readable medium of Claim 1, further comprising a validation component configured to validate the word-processor document.
3. The computer-readable medium of Claim 1, further comprising a fourth component for displaying the word-processor document.
4. The computer-readable medium of Claim 2, wherein the XSD represents a word-processor's rich formatting.
5. The computer-readable medium of Claim 4, wherein the XSD is published and is available to applications other than the word-processor.
6. The computer-readable medium of Claim 4, wherein the word-processor document includes hints to application that understand XML.

7. The computer-readable medium of Claim 5, wherein the action may be selected from parsing, modifying, reading, and creating the word-processor document.
8. The computer-readable medium of Claim 5, wherein the action may be fully recreating the word-processor document according to word processor's set of features.
9. The computer-readable medium of Claim 8, wherein the action may be storing an image within the word-processor as a binary encoding.
10. The computer-readable medium of Claim 9, wherein the action may be storing template information as a binary encoding with the word-processor document.
11. A method for handling a word-processing document, comprising:
  - parsing the document, wherein the document is contained within a single XML file and includes all the instructions necessary to display the document according to how a word-processor would display the document; and interpreting the document according to an XSD.
12. the method of Claim 11, further comprising displaying the document according to the instruction contained within the single XML file.
13. The method of Claim 12, further comprising modifying the document so as to conform with an XSD file.
14. The method of Claim 13, wherein the XSD includes definitions for all of the features incorporated within the word-processor.
15. The method of Claim 12, wherein displaying the document further comprises formatting the text according to style and properties contained within the single XML file.
16. The method of Claim 11, further comprising extracting text from the single XML file by searching for a single tag indicating that the information following the tag is text.
17. A system for creating, interpreting, and modifying a word-processor document stored as a single WPML file, comprising:
  - a WPML file;
  - a validation engine configured to validate the WPML file; and

a word processor configured to read a WPML file created in accordance with a schema.

18. The system of Claim 17, wherein the word processor is further configured to modify the WPML file in accordance with the schema.
  19. The system of Claim 17, wherein the word processor is further configured to output document to a display.
  20. The system of Claim 19, wherein the schema is published and is available to applications other than the word-processor.
  21. The system of Claim 19, wherein the WPML file includes hints to applications that understand XML.
  22. The system of Claim 19, wherein the word processor is further configured to create a document according to the schema.
  23. A method for handling a word-processing documents, substantially as herein describe with reference to the accompanying figures.
  24. A system for creating, interpreting, and modifying a word-processor document stored as a single WPML file, substantially as herein described with reference to the accompanying figures.
11. The Opponent puts at issue the invention, in so far as it is claimed to be:
1. Unpublished in New Zealand prior to the priority date of the claim;
  2. Unused in New Zealand prior to the priority date of the claim;
  3. Not obvious, both by publication and by use;
  4. An invention within the meaning of this Act;
  5. Sufficient.
12. The Opponent will contend that all of the Claims (1 – 24) are deficient in respect to one or more of the above issues.

## **GROUND 1 – PRIOR PUBLICATION**

### **s 21(1)(b)**

13. The Opponent will contend that the invention, so far as claimed in any claim of the complete specification, has been published in New Zealand before the priority date of the claim. In particular, the invention is prior published in the following publications, each of which it is submitted was available in New Zealand from a date prior to the earliest priority date to which the claims of the Subject Application are entitled.

#### **Document 5:**

14. AbiWord 2000 XSD – XML schema file – re Appendix 5

14.1. The AbiWord XSD file is an XML schema which is used to validate AbiWord XML word processing documents. It is directly comparable in function to the example Word Processing XSD file of the Subject Application.

14.2. A portion of the XSD, shows it was last modified on the 27<sup>th</sup> of April 2000 and that its purpose is the validation of AbiWord XML documents:

```
<xsd:annotation>
  <xsd:documentation>
    XML Schema for AbiWord Markup Language
    This should properly validate all AbiWord documents
    For validation, use http://cgi.w3.org/cgi-bin/xmlschema-check
    Please report errors to sam@uchicago.edu
    Modified on 4-27-00
  </xsd:documentation>
</xsd:annotation>
```

14.3. The AbiWord XSD document makes specific references to styles on line 25, sections on line 27, and data types on line 28. Further, this XSD could be processed using the same validation software as the example Word Processing XSD file of the Subject Application.

#### **Document 11:**

15. Docbook DTD – XML schema file – re Appendix 11

15.1. The Docbook DTD file is a XML schema used to validate Docbook documents. Docbook is used by authors to write manuscripts for publishers. It duplicates the functionality of the example Word XSD, except that it uses an XSD to validate instead of a DTD validator, and is so comparable. The primary DTD file, called “docbook.dtd”, contains links to other files which define chapters, sections and paragraphs of

books. It also defines page layouts, fonts and other document meta data.

**Document 8:**

16.OpenOffice.org DTD – XML schema file – re Appendix 8

16.1.The OpenOffice DTD file is a XML schema used to validate OpenOffice documents. It is directly comparable in function to XSD file in the Subject Application. The DTD is a collection of files, each separate file defining a separate aspect of the file format. The primary file (open\_office\_ref\_8.dtd) contains links to Styles on line 12, Text on line 20, and Charts on line 24. The DTD is an embodiment of the OpenOffice.org XML File Format specification, in that the DTD can validate an OpenOffice document to ensure it follows the specification.

**Document 13:**

17.OpenOffice 1.0 Application – re Appendix 13

17.1.The OpenOffice software application provides a Word Processor, Spreadsheet, and Presentation all in a single application. As discussed above there is a formal file specification, as published in 2000 and a XML schema which is used to validate documents. These elements are used in the OpenOffice application itself to write and read OpenOffice XML documents. The OpenOffice 1.0 application was published via Internet on May 1 2002. The OpenOffice application performs actions, displays and modifies XML documents as per the claims 1 through 16 of the patent application.

**Document 14:**

18.AbiWord Application – re Appendix 14

18.1.The AbiWord word processing application was first released in 1998. Thereafter it was included with many popular Linux distributions, including Red Hat 7.3 (attached and at Appendix 19). The AbiWord application stores its word processing files as a single XML file. AbiWord has DTD and XSD XML schema that are used to validate the word processing files. The AbiWord application performs actions, displays and modifies XML documents as per the claims 1 through 16 of the patent application.

**Document 15:**

19.KWord Application – re Appendix 15

19.1.KWord is a word processing application is a word processing application that has been copyright and available since 1998, and comes with a majority of Linux distributions. As an example, it was provided with Red Hat 7.3 in May 2002. KWord uses XML to store, display and modify it's word processing documents as per the claims 1 through 16 in the patent application.

**GROUND 2 – PRIOR USE**

**s.21(1)(d)**

20.Evidence will be adduced as to the New Zealand publication dates of the Documents 3, 7 and 12.

21.The Opponent will contend that the invention so far as claimed in any claim of the complete specification of the Subject Application was used in New Zealand before the priority date of any such claim.

22.The Opponent will adduce evidence as to the following:

Use by the Opponent

23.The Opponent used the following documents, either by way of communication of the contents of the documents to others or by way of receipt of communication of the contents of the documents from others, in New Zealand prior to the earliest priority date to which the Subject Application is entitled;

**Document 3:**

24.AbiWord 1.1 April 2002 XML Document – re Appendix 3

24.1.A word processing program AbiWord at least uses a single XML file to store data, and has been used in New Zealand in that way since 1998. AbiWord also uses an XML schema to validate and act on XML files;

24.2.To demonstrate use prior to the priority date, the Opponent includes an example of an AbiWord Document prepared prior to the priority date, 28<sup>th</sup> June 2002, at Appendix 3.

**Document 7:**

25. OpenOffice Version 1 XML File – re Appendix 7

25.1. OpenOffice incorporates a word processing program, and has done so in New Zealand since 2000. The OpenOffice word processor is at least capable of using XML as the file format for its documents, and their contents, it uses an XML schema to validate the XML files, and enables action to be undertaken on the XML file;

25.2. To demonstrate use prior to the priority date, the Opponent includes an example of an OpenOffice Document prepared prior to the priority date of the Subject Application, at Appendix 7.

**Document 12:**

26. Docbook XML Document file – re Appendix 12

26.1. The Docbook is an XML transform which translates a form of word processor markup into XML, for subsequent conversion into PDF documents. Docbook at least stores word processor documents as single XML files, uses an XML schema to validate the XML file, and then enables action on the XML file (i.e. conversion to a PDF file) to be undertaken, and included this functionality prior to the earliest priority date to which the Subject Application is entitled.

26.2. To demonstrate use prior to the priority date, the Opponent includes an example of an Docbook XML document prepared prior to the priority date at Appendix 12.

**GROUND 3 – OBVIOUSNESS – BY PUBLICATION**

**s 21(1)(e)**

27. The Opponent contends that:

27.1. The invention, so far as claimed in any claim of the Subject Application, is obvious and clearly does not involve any inventive step in regard to each, some or all the documents listed under Ground 1 of this Opposition and the additional documents cited under this ground, each of which it is submitted was available in New Zealand from a date prior to the earliest priority date of the claims of the Subject Application;

27.2. And also, having regard to what is common general knowledge in the field of the invention, which common general knowledge can be established from the following documents.

**Document 1:**

28. Extensible Markup Language (XML) 1.0 Specification (XML Specification) – re Appendix 1

28.1. This document discloses at least:

28.2. That using XML to represent word processing documents is merely the use of an open standard which was specifically designed to be able to represent documents. The XML Specification enumerates the purpose of the standard at paragraph 1.1, page 1:

The design goals for XML are:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

28.3. XML is not limited to word processing documents. It is clear from the above that using XML for word processing would have been an obvious and intended use of the specification. Word processing is merely a subset of the possible uses for XML.

**Document 2:**

29. XML Schema Part 1: Structures Second Edition (Schema Specification) – re Appendix 2

30. The Opponent contends that:

30.1. The Subject Application is merely the application of an existing standards to an obvious and intended use. It is clear from the Abstract of the Schema Specification that the use by the Subject Application of an XSD to validate an XML file is merely the use of an open standard as designed:

XML Schema: Structures specifies the XML Schema definition language, which offers facilities for describing the structure and constraining the contents of XML 1.0 documents, including those which exploit the XML Namespace facility. The schema language, which is itself represented in XML 1.0 and uses namespaces, substantially reconstructs and considerably extends the capabilities found in XML 1.0 document type definitions (DTDs). This specification depends on XML Schema Part 2: Datatypes.

30.2. This section specifies the extension of DTD functionality by the XSD XML Schema. The use of XSD schema in place of DTD schema in future word processing applications is an obvious consequence.

**Document 4:**

31. AbiWord 2001 DTD – XML schema file – re Appendix 4

31.1. The AbiWord DTD is an XML schema used to validate AbiWord XML word processing documents. It is directly comparable in function to the example XSD of the Subject Application.

31.2. The document provides reference to document styles, sections, and data. In particular at line 18.

**Document 10:**

32. KWord Version 1 DTD – XML schema file – re Appendix 10

32.1. The KWord DTD shows evidence that XML schema were used by word processing application for word processing documents prior to the priority date of the Subject Application:

```
<!-- $Id: kword.dtd,v 1.1 1998/12/15 22:37:27 kalle Exp $
This is an XML document type definition (DTD) for the KWord document format.
Written by Kalle Dalheimer <kalle@kde.org> with (obviously) input from
KWord's author Reginald Stadlbauer <reggie@kde.org>.
Please report all bugs to Kalle and Reginald at the above addresses.
-->
```

32.2. KWord used DTD schema for validation prior to the introduction of the XSD XML Schema standard.

32.3. The use of XSD by the Subject Application is the obvious consequence of the availability of a new tool intended for a demonstrably old purpose – validation. A developer skilled in the art in 2002 would have know of XSD schema; it would have been the contemporary technology to use for the purpose of validation. There is no inventive step.

32.4. The Opponent contends the development of an XSD for AbiWord, following the introduction of XSD standards, demonstrates the obvious and intended development of the art. Consequently, both DTD and XSD schema files exist for the AbiWord file format.

#### **GROUND 4 – OBVIOUSNESS**

##### **s 21(1)(e)**

###### US Patent Number 187060

33. Further, it has come to the attention of the Opponent that the US Patent Application on which the Subject Application relies for its priority date, and which is substantially the same as the Subject Application, has been rejected (4<sup>th</sup> April, 2005) by the United States Patent and Trademark Office. The grounds for rejection fall substantially under obviousness. The Opponent submits that the rejection of US Patent Number 187060 provides strongly persuasive authority for the rejection of the Subject Application on the same or similar grounds to the rejection of US Patent Number 187060 and that the Commissioner take notice of the rejection of US Patent Number 187060.

34. The US Patent and Trademark Office rejection (4<sup>th</sup> April, 2005) of US Patent Application Number 187060 is submitted as evidence at Appendix 18

#### **GROUND 5 – NOT AN INVENTION**

##### **s 21(1)(f)**

35. The Opponent will aver that the claimed invention is not an invention within the meaning of the Patents Act 1953

36. The Opponent repeats the description of prior art under Grounds 1 to 4 herein.

37. The Opponent submits that the following aspects of technology to which the alleged invention relates are well known and as a result the claimed invention can not be regarded as new at the earliest priority date available to any claim of the Subject Application:

37.1.The example XML Schema of the Subject Application merely provides for the identification of a the XML document;

37.2.The Subject Application provides no details to make a document, nor how to use the specification to correctly display or print a document;

37.3.The Subject Application does not claim a new manner of manufacture.

38.The Opponent therefore contends that the subject matter and disclosures in the Subject Application cannot in any way be regarded as new and do not comprise a contribution to the art or the the stock of human knowledge.

39.The Opponent further submits that the specification of the Subject Application is merely a recitation or collocation of common general knowledge which has been well know and publicly applied prior to the earliest priority date to which any of the claims of the Subject Application are entitled.

## **GROUND 6 – INSUFFICIENCY**

### **s 21(1)(g)**

40.The Opponent contents that all (or some) of the claims of the Subject Application are broad and indeterminate, and of a speculative character, in that the complete specification does not provide sufficiently detailed and full description in support thereof.

41.The Opponent submits that the specification of the Subject Application is lacking in detail specifically relating to:

41.1.Instruction on how such a document might be made;

41.2.The example XSD file is arbitrary and does not contain sufficient information for a software developer to implement the claimed invention;

41.3.The Subject Application provides merely a description of a word processing file and of a file to validate that word processing file – no information is provided about any unique method of making the file;

41.4.Consequently, there is a failure to disclose any process in the Subject Application.

42. Therefore, the Applicant has not fulfilled its obligations to provide a sufficient and fair description of the invention and the best method by which it is to be performed.

43. Accordingly, the Opponent contends that the specification of the Subject Application does not sufficiently and clearly identify the inventive subject matter of the Subject Application over the common general knowledge and the prior art. It is contended that this can only be indicative of an insufficiency of description of the invention, if such can possibly be claimed, and the means by which it is to be performed.

### **RELIEF SOUGHT**

44. Accordingly the New Zealand Open Source Society seeks the following relief:

44.1. Refusal of the New Zealand patent 525484;

44.2. Costs; and

44.3. Such other relief as the Commissioner deems appropriate.

45. The Opponent has drawn to the attention of the Applicant the deficiencies as set out in the Notice of Opposition, nonetheless the Applicant has persisted with the Subject Application.

Our address for service is JAMES & WELLS, Intellectual Property of Level 9, James & Wells Tower, 56 Cawley Street, Ellerslie, Private Bag 11907, DX CP34005, Auckland, New Zealand.

Dated \_\_\_\_\_ June 2005

**NEW ZEALAND OPEN SOURCE SOCIETY (INC)**  
**by its Attorneys**  
**JAMES & WELLS**

TO: The Commissioner  
The Intellectual Property Office of New Zealand  
LOWER HUTT

AND TO: The Applicant, by its attorneys AJ Park

## **ATTACHMENTS**

1. Copies of Red Hat Linux 7.3 on 5 Compact Disks (photocopies of Red Hat Linux 7.3 documentation at Appendix 19).
2. Software copies of documentary evidence provided in appendices following.

## **APPENDIX 1**

Extensible Markup Language (XML) 1.0 Specification published via the Internet  
(the full document is provided with software copies of documentary evidence,  
Attachment 2) page 1 following:



## **APPENDIX 2**

XML Schema Part 1: Structures Second Edition published via the Internet (the full document is provided with software copies of documentary evidence, Attachment 2) page 1 following:



## XML Schema Part 1: Structures

### W3C Recommendation 2 May 2001

**This version:**

<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>  
(in [XML](#) (with its own [DTD](#), [XSL stylesheet](#)) and [HTML](#)), with separate provision of the [schema](#) and [DTD](#) for schemas described herein.

**Latest version:**

<http://www.w3.org/TR/xmlschema-1/>

**Previous version:**

<http://www.w3.org/TR/2001/PR-xmlschema-1-20010330/>

**Editors:**

Henry S. Thompson (University of Edinburgh) [<ht@cogsci.ed.ac.uk>](mailto:ht@cogsci.ed.ac.uk)  
David Beech (Oracle Corporation) [<David.Beech@oracle.com>](mailto:David.Beech@oracle.com)  
Murray Maloney (for Commerce One) [<murray@muzmo.com>](mailto:murray@muzmo.com)  
Noah Mendelsohn (Lotus Development Corporation)  
[<Noah\\_Mendelsohn@lotus.com>](mailto:Noah_Mendelsohn@lotus.com)

[Copyright](#) ©2001 [W3C](#)<sup>®</sup> ([MIT](#), [INRIA](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#) and [software licensing](#) rules apply.

---

### Abstract

*XML Schema: Structures* specifies the XML Schema definition language, which offers facilities for describing the structure and constraining the contents of XML 1.0 documents, including those which exploit the XML Namespace facility. The schema language, which is itself represented in XML 1.0 and uses namespaces, substantially reconstructs and considerably extends the capabilities found in XML 1.0 document type definitions (DTDs). This specification depends on *XML Schema Part 2: Datatypes*.

### Status of this document

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.*

## **APPENDIX 3**

Example of an AbiWord Document available with Red Hat Linux 7.3  
(AbiWord\_SorceOverview.abw, dated 13<sup>th</sup> April 1999) following:

```

<!-- ===== -->
<!-- This file is an AbiWord document. -->
<!-- AbiWord is a free, Open Source word processor. -->
<!-- You may obtain more information about AbiWord at www.abisource.com -->
<!-- You should not edit this file by hand. -->
<!-- ===== -->

<!-- Build_ID = (none) -->
<!-- Build_Version = (none) -->
<!-- Build_Options = Debug:On -->
<!-- Build_Target = WIN32_4.0_i386_DBG -->
<!-- Build_CompilTime = 12:30:35 -->
<!-- Build_CompilDate = Apr 13 1999 -->

<AbiWord version="">
<section>
<p><c props="font-size:24pt">This document is obsolete as of the tree reorg which occurred
between 0.5.3 and 0.5.4. It will be revised as soon as possible.</c></p>
<p></p>
<p><c props="font-family:courier; font-size:11pt">$Id: AbiWord_SourceOverview.abw,v 1.8
1999/04/13 19:08:03 eric Exp $</c></p>
<p></p>
<p props="text-align:center"><c props="font-size:24pt; font-weight:bold">AbiWord Source
Overview</c></p>
<p props="text-align:center"><c props="font-size:11pt">Copyright (C) 1998-1999 AbiSource,
Inc. All Rights Reserved.</c></p>
<p props="text-align:center"></p>
<p props="text-align:center"><c props="font-style:italic">Jeff Hostetler</c></p>
<p props="text-align:center"><c props="font-family:courier">jeff@abisource.com</c></p>
<p props="text-align:center">AbiSource, Inc.</p>
<p props="text-align:center"></p>
<p props="text-align:center"><c props="font-family:courier">$Date: 1999/04/13 19:08:03
$/c></p>
<p></p>
<p props="margin-top:12pt"><c props="font-weight:bold">1. Introduction</c></p>
<p props="margin-top:12pt">This document gives an overview of the AbiWord source code. It
is intended as a high level overview of the organization and the architecture of the
AbiWord application and an overview of the cross-application framework that will be used as
a basis for other applications in the AbiSuite family. Specific details on each portion of
the source code will be given in other documents.</p>
<p props="margin-top:12pt">AbiWord is an GUI application written in C++. It currently runs
on Unix systems using X11 and the GTK toolkit and on Windows ('95, '98, NT) using the Win32
API. Ports are currently underway for BeOS and for the MacIntosh. A lot of time and
effort has been spent to make the source code as cross-platform as possible. As of build
0.3.0, our code is approximately 90% cross-platform (~46.5k cross-platform and ~6.5k
platform-specific). We have also spent time to abstract the application framework so that
we may share it with other AbiSuite applications. As of build 0.3.0, our code is
approximately 48% cross-application (~28.5k cross-application and ~31k AbiWord-specific);
this ratio will decrease as we add features to AbiWord, but it is a very good start.</p>
<p props="margin-top:12pt">This approach gives us great feature/bug consistency between
platforms. It also allows us to use native GUI libraries so that we get a native look-and-
feel on each platform. It is our hope that by extensively leveraging cross-platform and
cross-application code that we can minimize the effort needed to port the code to another
OS or GUI toolkit. It will also allow us to better leverage our QA and documentation
efforts.</p>
<p props="margin-top:12pt"><c props="font-weight:bold">2. Source Naming
Conventions</c></p>
<p props="margin-top:12pt">We have divided the source up into functional modules. Each
module is given its own subdirectory in the source tree. </p>
<p props="margin-top:12pt">Routines or classes which are considered <c props="font-
style:italic">public</c>(accessible outside the boundaries of a module) are given
uppercase prefixes; routines or classes considered module-private are given lowercase
prefixes. File and class prefixes are chosen to help reflect the module that contains it.
</p>
<p props="margin-top:12pt">Each module is responsible for compiling its own source and
building the appropriate library. Within a module the the platform-independent code is kept
in the <c props="font-family:courier">/xp</c> directory; the unix-specific code is kept in
the <c props="font-family:courier">/unix</c> directory; and the win32-specific code is kept
in the <c props="font-family:courier">/win</c> directory. If a module contains only
platform-independent code, it will only have an <c props="font-family:courier">/xp</c>
directory.</p>
<p props="margin-top:12pt">Most of the word-processor-core functionality is 100% cross-
platform code.</p>
<p props="margin-top:12pt">Most of the user-interface code is part cross-platform and part
platform-specific. In most cases, we create base classes in cross-platform code that
attempt to do as much as possible in cross-platform code and that also define interfaces
for functions that must be platform-specific. Each platform can then derive sub-classes and
'fill in the pieces'.</p>
<p props="margin-top:12pt"><c props="font-weight:bold">3. Source Directory
Organization</c></p>
<p props="margin-top:12pt"><c props="font-weight:bold">3.1. Cross-Application
Framework</c></p>
<p props="margin-top:0pt"></p>
<p props="margin-top:0pt">abi/src/xap {/xp, /unix, /win } -- These directories contain the
cross-application framework. These contain the classes to represent the application-
independent portion of the main application, top-level window frames, window lists, command
line argument handling, and various application-independent dialogs.</p>
<p props="margin-top:0pt"></p>

```

abi/src/ev { /xp, /unix, /win } -- These directories contain the code for the keyboard, mouse, menu, and toolbar event handlers. These provide the mechanism to implement the event-binding mechanism.

abi/src/ver { /xp } -- This directory defines code to internalize compile-time build information into the application. Build date and version and various compiler options are placed in various application variables.

**3.2. AbiWord-Specific**

abi/wp/ap { /xp, /unix, /win } -- These directories contain the AbiWord-Specific portion of the application-framework. These contain classes derived from the abi/src/xap classes. They contain the state tables used by the event-binding mechanism in abi/src/ev. They also contain the code for application-specific dialogs and the ruler widgets.

abi/wp/fmt { /xp } -- This directory contains the layout engine.

abi/wp/gr { /xp, /unix, /win } -- These directories contain the graphics classes used to draw to the screen. This includes basic drawing primitives as well as font operations.

abi/wp/impexp { /xp } -- This directory contains the Import and Export code used to read and write files in the various formats that we support.

abi/wp/main { /unix, /win } -- These directories contain the main program. These are stub entry points to call to package up the command line arguments and pass control to the application framework class.

abi/wp/ptbl { /xp } -- This directory contains the Piece Table code. This is the (quite complex) data structure used to represent the document in memory with complete undo/redo information.

**3.3. Miscellaneous Application-Independent**

abi/src/config -- This directory contains the Makefile declarations and rules used to build everything; all of the Makefiles in the source tree simply define their targets and include the font-family:courier.mk files found here. Additionally, the /platforms sub-directory contains platform-specific declarations.

abi/src/other -- This directory contains other (3rd party) source code that we have chosen to incorporate into our source. Currently this directory contains:

- expat -- an XML parser from James Clark. (This will be moving out of the /abi/src/other source tree and into a peer of /abi in the near future.)
- ispell -- the ISPELL spell-checker.
- abi/src/util { /xp, /unix, /win } -- These directories contain various utility routines and simple data structures that we have found useful. Some of these are used to isolate platform-specific behavior (such as string routines, debug messages, and assert macros) and others are used to provide useful data structures (such as hash tables, vectors, and string pools). These directories also contain various data conversion routines.

It is one of our goals to restrict platform-specific #ifdef's to code in the abi/src/util/xp directory.

**3.4. Additional Sources**

AbiWord also makes use of the following libraries:

- GTK and GLIB (on Unix)
- zlib and libpng

**4. Class Hierarchies**

**4.1. Application**

The Application class contains all of the application instance data (stuff that could be considered to be global).

- AP\_App (defined in abi/src/xap/xp/xap\_App.h)
- AP\_UnixApp (defined in abi/src/xap/unix/xap\_UnixApp.h)
- AP\_Win32App (defined in abi/src/xap/win/xap\_Win32App.h)

AP\_App manages information needed the application as a whole. This includes: application build and version information, the command-line arguments, a list of all open frames (documents and windows), pointers to the tables of possible menu and toolbar actions, pointers to the edit bindings. It also contains numerous abstract virtual methods for platform-specific things.

AP\_{Unix,Win32}App manages platform-specific application information. This includes: the Dialog Factory for application-persistent dialogs, the Toolbar Combo Factory, and a handle to the system clipboard.

Only one instance of this class is instantiated.

**4.2. Frame**

The Frame class contains all of the ...[TO BE CONTINUED...]

**5. Flow of Control**

main()

Execution begins with the platform-specific main entry point (i.e., WinMain(), or whatever) in /abi/src/wp/main/platform. This code is an ultra-minimal function that is responsible for capturing and command-line arguments and creating an instance of class AP\_{Unix,Win32}App.

```
<p props="margin-top:12pt">class AP_{Unix,Win32}App</p>
<p props="margin-left:0.5000in; margin-top:12pt">This class is responsible for initializing
any application resources (such as registering window classes, loading toolbar icons,
loading font directories, and connecting to the display).</p>
<p props="margin-left:0.5000in; margin-top:12pt">It also creates the first Frame and loads
the document named on the command-line or creates a new, untitled document.</p>
<p props="margin-left:0.5000in; margin-top:12pt"></p>
<p props="margin-top:12pt">[TO BE CONTINUED...]</p>
<p props="margin-top:12pt"></p>
</section>
</AbiWord>
```

## **APPENDIX 4**

Example of an AbiWord DTD – XML schema file published via the Internet  
(awml\_ref\_4.dtd, dated 3<sup>rd</sup> January 2000) following:

```

<!--
    AbiWord Markup Language DTD
    Current as of 3-01-00

    Currently, this document is intend for reference purposes only,
    and should not be interpreted as an absolute guide to the
    AbiWord file format.  A more comprehensive view can be gained
    from the code in files ie_exp_AbiWord_1.cpp and
    ie_imp_AbiWord_1.cpp These are currently definitve.  Other
    documentation is available in docs/AbiWord_DocumentFormat.abw

    Sam Tobin-Hochstadt
-->

<!ELEMENT abiword (styles?, section+, data?)>
<!ATTLIST abiword
    version CDATA "unnumbered">
<!-- This is the root element, with an attribute
    representing the version number. -->

<!ELEMENT styles (s*)>
<!ELEMENT s EMPTY>
<!ATTLIST s
    basedon CDATA #IMPLIED
    name CDATA #REQUIRED
    type CDATA #IMPLIED
    props CDATA #IMPLIED>
<!-- Styles allow for paragraph level formatting.
    The basedon attribute specifies inheritance
    and the props attribute specifies features
    of the style. -->

<!ELEMENT section (p+)>
<!ATTLIST section
    props CDATA #IMPLIED
    type (footer | CDATA) #IMPLIED
    id ID #IMPLIED
    footer CDATA #IMPLIED>
<!-- Sections are collections of paragraphs.
    They may specify footers to be attached,
    or formatting properties. -->

<!ELEMENT p (#PCDATA | c | field | image | cbr | pbr | br)*>
<!ATTLIST p
    props CDATA #IMPLIED
    level (0|1|2|3|4|5|6|7|8|9|10 | CDATA) #IMPLIED
    style CDATA #IMPLIED
    id ID #IMPLIED>
<!-- p element contain text, images, fields or character
    spans.  Level is used for lists, style for styles
    and props for formatting. -->

<!ELEMENT c (#PCDATA)>
<!ATTLIST c
    props CDATA #IMPLIED
    type (list-label | CDATA) #IMPLIED
    style CDATA #IMPLIED>
<!-- c is used to specify character based formatting
    which is done with the props attribute.
    type is used for lists. style can specify
    predefined styles. -->

<!ELEMENT field EMPTY>
<!ATTLIST field
    type (list-label | page-number | page-count | time | CDATA) #REQUIRED>
<!-- fields represent calculated data on the page.  This format is currently
    in flux. -->

<!ELEMENT image EMPTY>
<!ATTLIST image
    dataid CDATA #REQUIRED
    props CDATA #IMPLIED>
<!-- image refers to an image in
    d section.  it is used much
    the same as in HTML. -->

<!ELEMENT br EMPTY>
<!ELEMENT pbr EMPTY>
<!ELEMENT cbr EMPTY>
<!-- These represent line,
    page and column breaks. -->

<!ELEMENT data (d+)>
<!ELEMENT d (#PCDATA)>
<!ATTLIST d
    name CDATA #REQUIRED>
<!-- d is used to store actual

```

data, which currently means  
images. -->

## **APPENDIX 5**

Example of an AbiWord 2001 XSD – XML schema file published via the Internet (awml\_ref\_5.xsd, dated 27<sup>th</sup> April 2000) following:

```

<?xml version="1.0"?>

<xsd:schema targetNamespace="urn:abisource:awml"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2000/10/XMLSchema
    http://www.w3.org/2000/10/XMLSchema.xsd">

  <xsd:annotation>
    <xsd:documentation>
      XML Schema for AbiWord Markup Language
      This should properly validate all AbiWord documents
      For validation, use http://cgi.w3.org/cgi-bin/xmlschema-check
      Please report errors to sam@uchicago.edu
      Modified on 4-27-00
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="abiword" type="abiwordType"/>

  <xsd:complexType name="abiwordType">
    <xsd:sequence>
      <xsd:element name="styles" type="stylesType" minOccurs="0"/>
      <xsd:element name="lists" type="listsType" minOccurs="0"/>
      <xsd:element name="section" type="sectionType" maxOccurs="unbounded"/>
      <xsd:element name="data" type="dataType" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="xsd:string" use="default" value="unnumbered"/>
  </xsd:complexType>

  <xsd:complexType name="stylesType">
    <xsd:sequence>
      <xsd:element name="s" maxOccurs="unbounded" minOccurs="0">
        <xsd:complexType>
          <xsd:attribute name="basedon" type="xsd:string"/>
          <xsd:attribute name="name" type="xsd:string" use="required"/>
          <xsd:attribute name="type" type="xsd:string"/>
          <xsd:attribute name="props" type="PropsAtt"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="dataType">
    <xsd:sequence>
      <xsd:element name="d" maxOccurs="unbounded" minOccurs="1">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="name" type="xsd:string"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="listsType">
    <xsd:sequence>
      <xsd:element name="l" maxOccurs="unbounded" minOccurs="1">
        <xsd:complexType>
          <xsd:attribute name="id" type="xsd:ID"/>
          <xsd:attribute name="parentid" type="xsd:ID"/>
          <xsd:attribute name="type" type="xsd:nonNegativeInteger"/>
          <xsd:attribute name="startvalue" type="xsd:integer"/>
          <xsd:attribute name="format" type="xsd:string"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="sectionType">
    <xsd:sequence>
      <xsd:element name="p" type="pType" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="props" type="PropsAtt"/>
    <xsd:attribute name="type" type="TypeAttS"/>
    <xsd:attribute name="id" type="xsd:ID"/>
    <xsd:attribute name="footer" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="pType">
    <xsd:sequence>
      <xsd:element name="c" minOccurs="0" maxOccurs="unbounded">

```

```

    <xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
    <xsd:attribute name="props" type="PropsAtt"/>
    <xsd:attribute name="type" type="TypeAttC"/>
    <xsd:attribute name="style" type="xsd:string"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

<xsd:element name="field" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:attribute name="type" type="TypeAttF" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="image" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:attribute name="dataid" type="xsd:string" use="required"/>
    <xsd:attribute name="props" type="PropsAtt"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="cbr" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType/>
</xsd:element>

<xsd:element name="pbr" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType/>
</xsd:element>

<xsd:element name="br" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType/>
</xsd:element>
</xsd:sequence>

<xsd:attribute name="props" type="xsd:string"/>
<xsd:attribute name="style" type="xsd:string"/>
<xsd:attribute name="id" type="xsd:ID"/>
<xsd:attribute name="level">
    <xsd:simpleType>
<xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:maxExclusive value="100"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>

<xsd:attribute name="listid">
    <xsd:simpleType>
    <xsd:restriction base="xsd:nonNegativeInteger"/>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="TypeAttF">
    <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:enumeration value="list-label"/>
    <xsd:enumeration value="time"/>
    <xsd:enumeration value="page-count"/>
    <xsd:enumeration value="page-number"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TypeAttC">
    <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:enumeration value="list-label"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TypeAttS">
    <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:enumeration value="footer"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="PropsAtt">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
</xsd:schema>

```

## **APPENDIX 6**

THIS PAGE DELETED.

## **APPENDIX 7**

Example of an OpenOffice Version 1 XML Document (OSSinForeignAffairs.sxw, dated 18<sup>th</sup> April 2002) following:

1150 East Coast Road,  
Redvale,  
RD2 Albany,

18 April 2002,

Phil Goff  
Minister of Foreign Affairs and Trade  
Parliament

Dear Mr Goff,

Recently there has been news that the Ministry of Foreign Affairs and Trade has decided to spend ten million dollars on new computers in order to bring them up to date, and to allow communication with more modern systems over the Internet. This is to be commended.

I am sending this letter and attached report to make you aware of a possibility which might save the Ministry hundreds of thousands of dollars in the initial purchase of computers and software, and substantial amounts of money on an ongoing basis.

The solution would provide a more reliable platform for your staff, and one which has been proven more secure than any comparable systems. In addition it would probably provide some very good public relation for the Government in the Information Technology industry, which would be good in an election year.

Please read the attached report for the details of the proposal. If you are interested in taking this proposal further we are prepared to meet you in person to discuss how we can help.

Yours Sincerely,

Peter Harrison

## **APPENDIX 8**

Example of an OpenOffice DTD – XML schema file published via the Internet  
(open\_office\_ref\_8.dtd, dated 18<sup>th</sup> September 2000) following:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  $Id: office.dtd,v 1.1 2000/09/18 17:07:00 hr Exp $

  Copyright 2000 Sun Microsystems Inc.
-->

<!ENTITY % dtypes-mod SYSTEM "dtypes.mod">
%dtypes-mod;
<!ENTITY % namespace-mod SYSTEM "namespace.mod">
%namespace-mod;
<!ENTITY % style-mod SYSTEM "style.mod">
%style-mod;
<!ENTITY % office-mod SYSTEM "office.mod">
%office-mod;
<!ENTITY % meta-mod SYSTEM "meta.mod">
%meta-mod;
<!ENTITY % drawing-mod SYSTEM "drawing.mod">
%drawing-mod;
<!ENTITY % text-mod SYSTEM "text.mod">
%text-mod;
<!ENTITY % table-mod SYSTEM "table.mod">
%table-mod;
<!ENTITY % chart-mod SYSTEM "chart.mod">
%chart-mod;
<!ENTITY % datastyl-mod SYSTEM "datastyl.mod">
%datastyl-mod;
```

## **APPENDIX 9**

THIS PAGE DELETED

## **APPENDIX 10**

Example of an KWord Version 1 DTD – XML schema file published via the Internet (kword\_ref\_10.dtd, dated 15<sup>th</sup> December 1998) following:

```

<!-- $Id: kword.dtd,v 1.1 1998/12/15 22:37:27 kalle Exp $

      This is an XML document type definition (DTD) for the KWord document format.
      Written by Kalle Dalheimer <kalle@kde.org> with (obviously) input from
      KWord's author Reginald Stadlbauer <reggie@kde.org>.

-->
      Please report all bugs to Kalle and Reginald at the above addresses.

-->

<!-- A KWord document consists of a paper description, a number of
      attributes and the framesets which contain the actual data. It
      can also define a number of styles.

      Attributes:

      author:      The document's author
      email:       The document's author's email address
      editor:      The program this file was written with
      mime:        The MIME type (must always have the value application/x-kword

-->
<!ELEMENT DOC (PAPER, ATTRIBUTES, FRAMESETS, STYLES)>
<!ATTLIST DOC  author CDATA #IMPLIED
               email CDATA #IMPLIED
               editor CDATA #IMPLIED
               mime CDATA "application/x-kword">

<!-- Describes the page format. Can have no or one PAPERBORDERS child that
      describes the margins.

      Attributes:

      format:      integer value for the page format
      width:       width in millimeters
      height:      height in millimeters
      orientation: 0 for landscape, 1 for portrait
      columns:     the number of columns (only in wordprocessing mode)
      columnspacing: the gap between the columns (only in wordprocessing mode)

-->
<!ELEMENT PAPER (PAPERBORDERS?)>
<!ATTLIST PAPER  format CDATA #REQUIRED
                 width CDATA #REQUIRED
                 height CDATA #REQUIRED
                 orientation CDATA #REQUIRED
                 columns CDATA #REQUIRED
                 columnspacing CDATA #REQUIRED>

<!-- Describes the margins of the page.

      Attributes:

      left:        left margin in pixels
      right:       right margin in pixels
      top:         top margin in pixels
      bottom:      bottom margin in pixels

-->
<!ELEMENT PAPERBORDERS EMPTY>
<!ATTLIST PAPERBORDERS left CDATA #REQUIRED
                      right CDATA #REQUIRED
                      top CDATA #REQUIRED
                      bottom CDATA #REQUIRED>

<!-- Contains general attributes of the document. Has no children.

      Attributes:

      processing:  0 for word processing, 1 for DTP
      standardpage: currently ignored

-->
<!ELEMENT ATTRIBUTES EMPTY>
<!ATTLIST ATTRIBUTES  processing CDATA #REQUIRED
                      standardpage CDATA #IMPLIED>

<!-- Container that as a number of FRAMESET children.

-->
<!ELEMENT FRAMESETS (FRAMESET+)>

<!-- One frameset that bundles a number of FRAME elements.

      Attributes:

      frametype:      1 for text, 2 for graphics
      autocreatenewframe: ???

-->

```

```

<!ELEMENT FRAMESET (FRAME|PARAGRAPH)*>
<!ATTLIST FRAMESET
    frameType CDATA #REQUIRED
    autoCreateNewFrame CDATA #REQUIRED>

<!ELEMENT FRAME EMPTY>
<!ATTLIST FRAME
    left CDATA #REQUIRED
    right CDATA #REQUIRED
    top CDATA #REQUIRED
    bottom CDATA #REQUIRED
    runaround CDATA #REQUIRED
    runaroundGap CDATA #REQUIRED>

<!ELEMENT PARAGRAPH (TEXT|FORMATS|LAYOUT)*>
<!ELEMENT TEXT EMPTY>
<!ATTLIST TEXT value CDATA #IMPLIED>
<!ELEMENT FORMATS (FORMAT*)>
<!ELEMENT FORMAT (COLOR|FONT|SIZE|WEIGHT|ITALIC|UNDERLINE|VERTALIGN)*>
<!ATTLIST FORMAT
    id CDATA #IMPLIED
    pos CDATA #IMPLIED
    len CDATA #IMPLIED>

<!ELEMENT COLOR EMPTY>
<!ATTLIST COLOR
    red CDATA #REQUIRED
    green CDATA #REQUIRED
    blue CDATA #REQUIRED>

<!ELEMENT FONT EMPTY>
<!ATTLIST FONT name CDATA #REQUIRED>
<!ELEMENT SIZE EMPTY>
<!ATTLIST SIZE value CDATA #REQUIRED>
<!ELEMENT WEIGHT EMPTY>
<!ATTLIST WEIGHT value CDATA #REQUIRED>
<!ELEMENT ITALIC EMPTY>
<!ATTLIST ITALIC value CDATA #REQUIRED>
<!ELEMENT UNDERLINE EMPTY>
<!ATTLIST UNDERLINE value CDATA #REQUIRED>
<!ELEMENT VERTALIGN EMPTY>
<!ATTLIST VERTALIGN value CDATA #REQUIRED>
<!ELEMENT LAYOUT
    (NAME|FOLLOWING|FLOW|OFFSETS|INDENTS|COUNTER|LINESPACING|
LEFTBORDER|
    RIGHTBORDER|TOPBORDER|BOTTOMBORDER|FORMAT|
TABULATOR)*>
<!ELEMENT NAME EMPTY>
<!ATTLIST NAME value CDATA #REQUIRED>
<!ELEMENT FOLLOWING EMPTY>
<!ATTLIST FOLLOWING name CDATA #REQUIRED>
<!ELEMENT FLOW EMPTY>
<!ATTLIST FLOW value CDATA #REQUIRED>
<!ELEMENT OFFSETS EMPTY>
<!ATTLIST OFFSETS
    head CDATA #REQUIRED
    foot CDATA #REQUIRED>
<!ELEMENT INDENTS EMPTY>
<!ATTLIST INDENTS
    first CDATA #REQUIRED
    left CDATA #REQUIRED>
<!ELEMENT COUNTER EMPTY>
<!ATTLIST COUNTER
    type CDATA #REQUIRED
    depth CDATA #REQUIRED
    bullet CDATA #REQUIRED
    start CDATA #REQUIRED
    numberingtype CDATA #REQUIRED
    lefttext CDATA #IMPLIED
    righttext CDATA #IMPLIED
    bulletfont CDATA #REQUIRED>
<!ELEMENT LINESPACING EMPTY>
<!ATTLIST LINESPACING value CDATA #REQUIRED>
<!ELEMENT LEFTBORDER EMPTY>
<!ATTLIST LEFTBORDER
    red CDATA #REQUIRED
    green CDATA #REQUIRED
    blue CDATA #REQUIRED
    style CDATA #REQUIRED
    width CDATA #REQUIRED>
<!ELEMENT RIGHTBORDER EMPTY>
<!ATTLIST RIGHTBORDER
    red CDATA #REQUIRED
    green CDATA #REQUIRED
    blue CDATA #REQUIRED
    style CDATA #REQUIRED
    width CDATA #REQUIRED>
<!ELEMENT TOPBORDER EMPTY>
<!ATTLIST TOPBORDER
    red CDATA #REQUIRED
    green CDATA #REQUIRED
    blue CDATA #REQUIRED
    style CDATA #REQUIRED
    width CDATA #REQUIRED>
<!ELEMENT BOTTOMBORDER EMPTY>
<!ATTLIST BOTTOMBORDER
    red CDATA #REQUIRED
    green CDATA #REQUIRED
    blue CDATA #REQUIRED
    style CDATA #REQUIRED
    width CDATA #REQUIRED>
<!ELEMENT TABULATOR EMPTY>
<!ATTLIST TABULATOR
    mmpos CDATA #REQUIRED
    ptpos CDATA #REQUIRED

```

```
type CDATA #REQUIRED>
<!ELEMENT STYLES (STYLE*)>
<!ELEMENT STYLE (NAME|FOLLOWING|FLOW|OFFSETS|INDENTS|COUNTER|LINE SPACING|LEFTBORDER|
RIGHTBORDER|
TOPBORDER|BOTTOMBORDER|FORMAT)*>
```

## **APPENDIX 11**

Example of a Docbook DTD – XML schema file available with Red Hat Linux 7.3 (docbook.dtd, dated 19<sup>th</sup> June 2000) following:

```

<!-- ..... -->
<!-- DocBook DTD V4.1 ..... -->
<!-- File docbook.dtd ..... -->

<!-- Copyright 1992-2000 HaL Computer Systems, Inc.,
O'Reilly & Associates, Inc., ArborText, Inc., Fujitsu Software
Corporation, and the Organization for the Advancement of
Structured Information Standards (OASIS).

$Id: docbook.dtd,v 1.11 2000/06/19 14:42:02 nwalsh Exp $

Permission to use, copy, modify and distribute the DocBook DTD and
its accompanying documentation for any purpose and without fee is
hereby granted in perpetuity, provided that the above copyright
notice and this paragraph appear in all copies. The copyright
holders make no representation about the suitability of the DTD for
any purpose. It is provided "as is" without expressed or implied
warranty.

If you modify the DocBook DTD in any way, except for declaring and
referencing additional sets of general entities and declaring
additional notations, label your DTD as a variant of DocBook. See
the maintenance documentation for more information.

Please direct all questions, bug reports, or suggestions for
changes to the docbook@lists.oasis-open.org mailing list. For more
information, see http://www.oasis-open.org/docbook/.
-->

<!-- ..... -->

<!-- This is the driver file for V4.1 of the DocBook DTD.
Please use the following formal public identifier to identify it:

"-//OASIS//DTD DocBook V4.1//EN"

For example, if your document's top-level element is Book, and
you are using DocBook directly, use the FPI in the DOCTYPE
declaration:

<!DOCTYPE Book PUBLIC "-//OASIS//DTD DocBook V4.1//EN" [...]>

Or, if you have a higher-level driver file that customizes DocBook,
use the FPI in the parameter entity declaration:

<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
%DocBookDTD;

The DocBook DTD is accompanied by an SGML declaration.

See the documentation for detailed information on the parameter
entity and module scheme used in DocBook, customizing DocBook and
planning for interchange, and changes made since the last release
of DocBook.
-->

<!-- ..... -->
<!-- Notation declarations ..... -->

<!ENTITY % dbnotn.module "INCLUDE">
<![ %dbnotn.module; [
<!ENTITY % dbnotn PUBLIC
"-//OASIS//ENTITIES DocBook Notations V4.1//EN">
%dbnotn;
<!--end of dbnotn.module-->]]>

<!-- ..... -->
<!-- ISO character entity sets ..... -->

<!ENTITY % dbcent.module "INCLUDE">
<![ %dbcent.module; [
<!ENTITY euro SDATA "[euro ]"><!-- euro sign, U+20AC NEW -->
<!ENTITY % dbcent PUBLIC
"-//OASIS//ENTITIES DocBook Character Entities V4.1//EN">
%dbcent;
<!--end of dbcent.module-->]]>

<!-- ..... -->
<!-- DTD modules ..... -->

<!-- Information pool ..... -->

<!ENTITY % dbpool.module "INCLUDE">
<![ %dbpool.module; [
<!ENTITY % dbpool PUBLIC
"-//OASIS//ELEMENTS DocBook Information Pool V4.1//EN">
%dbpool;
<!--end of dbpool.module-->]]>

<!-- Redeclaration placeholder ..... -->

```

```

<!ENTITY % intermod.redecl.module "IGNORE">
<![ %intermod.redecl.module; [
%rdbmods;
<!--end of intermod.redecl.module-->]]>

<!-- Document hierarchy ..... -->

<!ENTITY % dbhier.module "INCLUDE">
<![ %dbhier.module; [
<!ENTITY % dbhier PUBLIC
"-//OASIS//ELEMENTS DocBook Document Hierarchy V4.1//EN">
%dbhier;
<!--end of dbhier.module-->]]>

<!-- ..... -->
<!-- Other general entities ..... -->

<!ENTITY % dbgenent.module "INCLUDE">
<![ %dbgenent.module; [
<!ENTITY % dbgenent PUBLIC
"-//OASIS//ENTITIES DocBook Additional General Entities V4.1//EN">
%dbgenent;
<!--end of dbgenent.module-->]]>

<!-- End of DocBook DTD V4.1 ..... -->
<!-- ..... -->

```

## **APPENDIX 12**

Example of a Docbook XML Document available with Red Hat Linux 7.3 (documentation.xml, dated 6<sup>th</sup> January 2002) following:

```

<!DOCTYPE book
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd" [
<!ENTITY reference.xml SYSTEM "reference.xml" NDATA SGML>
<!ENTITY publishing.xml SYSTEM "publishing.xml">
<!ENTITY extensions.xml SYSTEM "extensions.xml">
<!ENTITY templates.xml SYSTEM "templates.xml">
<!ENTITY html-param.xml SYSTEM "html/param.xml" NDATA SGML>
<!ENTITY tools-profiling.xml SYSTEM "tools/profiling.xml" NDATA SGML>
<!ENTITY htmlhelp.xml SYSTEM "htmlhelp.xml">
]>
<book>
<bookinfo>
<title>DocBook XSL Stylesheet Documentation</title>
<!-- $Id: documentation.xml,v 1.5 2002/01/06 21:06:57 nwalsh Exp $ -->

<authorgroup>
<author><surname>Walsh</surname>
<firstname>Norman</firstname></author>
<othercredit><surname>Stayton</surname>
<firstname>Bob</firstname></othercredit>
<othercredit><surname>Kosek</surname>
<firstname>Ji&raron;&iacute;</firstname>
</othercredit>
</authorgroup>

<copyright><year>1999</year><year>2000</year><year>2001</year>
<holder>Norman Walsh</holder>
</copyright>
</bookinfo>

<preface><title>Preface</title>

<para>This is the beginnings of documentation for the DocBook XSL
Stylesheets.</para>

</preface>

&publishing.xml;
&extensions.xml;

<chapter>
<title>Tools documentation</title>

<para>There are several tools useful for usage with DocBook in
directory <filename>tools</filename>.</para>

<itemizedlist>
<listitem>
<para><ulink url="tools/profiling.html">Profiling</ulink></para>
</listitem>
</itemizedlist>

</chapter>

<chapter>
<title>Reference Documentation</title>

<para>Reference documentation is also available:</para>

<itemizedlist>
<listitem><para><olink type="title" localinfo="reference.html"
targetdocent="reference.xml"/></para>
</listitem>
</itemizedlist>
</chapter>

&templates.xml;
&htmlhelp.xml;
</book>

```

## **APPENDIX 13**

OpenOffice 1.0 Application was published via Internet on May 1 2002

## **APPENDIX 14**

AbiWord Application available with Red Hat Linux 7.3

## **APPENDIX 15**

KWord Application available with Red Hat Linux 7.3

## **APPENDIX 16**

THIS PAGE DELETED

## **APPENDIX 17**

Companies putting their name to this Statement of Case in Opposition to the Subject Application:

Catalyst Ltd  
PO Box 11-053  
Manners Street  
WELLINGTON

Egressive Ltd  
PO Box 24162  
CHRISTCHURCH

Nothing But Net Ltd  
Unit A6, 400 Rosedale Road  
Albany  
AUCKLAND

## **APPENDIX 18**

United States Patent and Trademark Office rejection of US Patent Application  
187060 following:

## **APPENDIX 19**

Photocopies of Red Hat Linux 7.3 documentation following:

# Linux

**Red Hat Linux features**

- KDE 3.0 • GNOME 1.4 • Evolution • GNOME Meeting • My Project • Mozilla • cups • postfix • kernel 2.4.18 • Apache 1.3.23-9 • gcc 2.96-104 • glibc 2.2.4 • Perl 5.6.1 • Emacs 21.1 • XFree86 4.2 • Xine • gPhoto • GIMP • New accessibility features

**Red Hat Linux basic service**

- 30 days access to Red Hat Network Basic Service
- gcc 3.1 delivered via Red Hat Network

**Other applications**

- StarOffice 5.2
- Sun's Java(TM) 2 Platform Standard Edition v.1.3.1
- Productivity applications CD

**The new features in Red Hat Linux Personal offer everything needed for a personal productivity workstation.**

- Easy** installation and upgrade program
- Build** a firewall at installation
- Dual-boot** with other operating systems
- Graphical** tools to set up services
- Use** KDE 3.0 and GNOME 1.4 desktops
- Mozilla** browser, Evolution email program, and GNOME Meeting Internet tools
- USB** support for multimedia equipment
- Full** StarOffice 5.2 Office Suite
- Manage** Red Hat Network Basic Service
- File** management with Nautilus

Copyright © 2002 Red Hat, Inc. All rights reserved. "Red Hat" and the Red Hat "Shadow Man" logo are registered trademarks and "Linux" is a trademark of Red Hat, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds, Sun, Sun Microsystems, the Sun Logo, StarOffice, the StarOffice logo and Sun are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries, and are used under license. This product includes software developed by the OpenOffice.org project. All other trademarks are the property of their respective holders.

The Red Hat Linux End User License Agreement can be seen online at <http://www.redhat.com/licenses>

W020223-01  
Revised/Updated August 2002  
Product manufactured in USA

ISBN 0-3847-40238-6  
UPC 1-48849-2504-6

Sum StarOffice fpm

# Linux

The Official Red Hat Linux Installation Guide is written to assist users with CD-ROM-based installations of Red Hat Linux 7.3. With helpful screenshots and important tips, this guide walks you step-by-step through the installation process.

Post-installation topics are covered in these guides:

- Official Red Hat Linux 7.3 Getting Started Guide
- Official Red Hat Linux 7.3 Customization Guide
- Official Red Hat Linux 7.3 Reference Guide

The Red Hat Network User Reference Guide can be found online at [rhn.redhat.com](http://rhn.redhat.com).

MAN0047AP  
Printed in Singapore.

Copyright © 2002 Red Hat, Inc. All rights reserved. "Red Hat" and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the US and other countries. Linux is a registered trademark of Linus Torvalds. All other trademarks shown are the property of their respective holders.



